

Penerapan metode *upper bound* pada *multi depot vehicle routing problem* (MDVRP)

Fahimatul Izzah¹, Mohamad Yasin², dan Sapti Wahyuningsih³
PROGRAM STUDI S1 MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS NEGERI MALANG
Email: iezhaq@gmail.com

Abstrak: *Multi Depot Vehicle Routing Problem*(MDVRP) merupakan permasalahan yang meliputi beberapa depot yang melayani beberapa *customer* dengan total jarak minimum dan kembali ke depot yang sama tanpa melanggar kendala kapasitas kendaraan. Terdapat tiga tahap dalam permasalahan MDVRP, diantaranya adalah pengelompokan, pembentukan rute, dan pengurutan rute. Pada tahap pengelompokan, beberapa *customer* dikelompokkan berdasarkan depot terdekat. Pada tahap pembentukan rute, pencarian rute menggunakan metode *Upper Bound*. Terdapat dua langkah pada metode tersebut yakni pemilihan depot sebagai titik awal pencarian rute dan dilanjutkan dengan pencarian jarak terpendek antar titik yang terpilih, apabila melanggar kapasitas maka pencarian rute langkah diulangi sampai semua titik termuat dalam suatu rute. Pada Tahap terakhir, urutan pengiriman dipilih sehingga urutan pengiriman dimulai dari depot ke *customer* terdekat, kemudian ke *customer* berikutnya yang terdekat dari *customer* sebelumnya, dan seterusnya. Untuk mengimplementasikan proses penyelesaian permasalahan MDVRP dengan metode *Upper Bound* maka dibuat program menggunakan *software Borland Delphi 07*.

Kata Kunci: *Multi Depot Vehicle Routing Problem* (MDVRP), Metode *Upper Bound*

Abstract: Multi Depot Vehicle Routing Problem (MDVRP) is a problem that includes several depots that serve the customer with a total minimum distance and return to the same depot without violating the capacity constraints of the vehicle. There are three stages in MDVRP problems, such as grouping, formation routes, and route sequencing. In the grouping stage, the customers grouped by the nearest depot. At the formation stage, route search using the Upper Bound. There are two steps in the method of selecting the depot as a starting point route search and continued to search the shortest distance between selected points, if the violation of the capacity then routing step is repeated until all the points contained in a route. At the last stage, the order of delivery is chosen so that the sequence begins delivery from the closest depot to the customer, then the next customer to the nearest of the previous customer, and so on. To implement the process of solving MDVRP problems with the Upper Bound method then created a program using Borland Delphi software 07.

Keywords: Multi Depot Vehicle Routing Problem (MDVRP), Upper Bound method

Teori graph merupakan salah satu cabang ilmu matematika yang memiliki kegunaan untuk memodelkan suatu permasalahan yang bertujuan agar dapat diselesaikan secara mudah. Dalam graph terdapat berbagai permasalahan yang dapat diterapkan pada kehidupan sehari-hari. Salah satu permasalahan dalam graph adalah *Multi Depot Vehicle Routing Problem* (MDVRP).

-
1. Fahimatul Izzah adalah mahasiswa jurusan Matematika FMIPA Universitas Negeri Malang
 2. Mohamad Yasin adalah dosen jurusan Matematika FMIPA Universitas Negeri Malang
 3. Sapti Wahyuningsih adalah dosen jurusan Matematika FMIPA Universitas Negeri Malang

Beberapa algoritma dan metode dari MDVRP yang telah dibahas. Salah satunya terdapat pada skripsi Masrurroh yaitu Algoritma *Clark and Wright* (Caccetta, 2013) yang contoh penerapannya menghasilkan 4 rute dan total jarak 797km untuk 2 depot dan 9 *customer* dan pada skripsi Prihartinie yaitu Metode *Insertion Heuristik* (Campbell, 2004) yang contoh penerapannya menghasilkan 4 rute dan total jarak 797km untuk 2 depot dan 9 *customer*.

Suatu shortest path yang melewati seluruh titik dalam suatu graph dan kembali ke titik awal disebut Travelling Salesman Problem (TSP). Suatu TSP menghasilkan cycle tunggal, perluasan suatu TSP yang menghasilkan cycle lebih dari satu disebut VRP. Terdapat beberapa algoritma dan metode untuk pencarian TSP. Salah satunya adalah metode Upper Bound, dimana hasil pencariannya merupakan sebuah cycle dengan jarak minimum.

Aldous (2000) dalam bukunya halaman 193 menuliskan bahwa berikut ini langkah - langkah metode *Upper Bound* pada *Travelling Salesman Problem*:

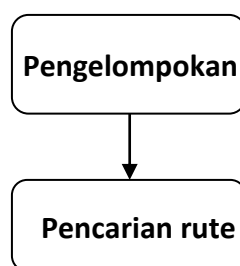
- START dengan himpunan titik hingga, dimana setiap titik terhubung oleh sisi berbobot.
- Langkah 1 : Pilih sebarang titik dan temukan titik yang dihubungkan dengan bobot sisi minimum. Gambar kedua titik tersebut dan hubungkan kedua titik dengan dua sisi membentuk *cycle*. Beri orientasi *cycle* searah jarum jam.
- Langkah 2 : Temukan titik yang belum tergambar dan hubungkan pada sebuah sisi berbobot tersedikit pada titik yang telah tergambar. Masukkan titik baru ini ke *cycle* di depan titik yang terdekat.

Ulangi Langkah 2 sampai semua titik terhubung dengan *cycle*, kemudian STOP. *Cycle* berbobot yang diperoleh adalah *Cycle* Hamilton dan bobot total diperoleh dari jumlah bobot pada setiap sisinya merupakan batas atas untuk suatu solusi pada *Travelling Salesman Problem*.

Berdasarkan uraian di atas akan dibahas tentang permasalahan MDVRP dengan menggunakan salah satu metode *shortest path* yakni *Upper Bound*. Pembahasan ini diharapkan dapat (1) mendeskripsikan tahapan untuk metode *Upper Bound* pada MDVRP, (2) menerapkan metode *Upper Bound* pada contoh penerapan MDVRP, (3) mengimplementasi metode *Upper Bound* ke dalam bahasa pemrograman *Borland Delphi 07*.

PEMBAHASAN

Multi Depot Vehicle Routing Problem (MDVRP) merupakan salah satu perluasan dari *Vehicle Routing Problem* (VRP) dimana beberapa kendaraan berangkat dari beberapa depot dan kembali ke depot asal sampai semua rute terlewati. Permasalahan MDVRP yaitu meminimalkan rute dengan memperhatikan kendala kapasitas kendaraan tanpa dipengaruhi kendala biaya dan waktu, dengan kata lain kendala biaya dan waktu tempuh diabaikan. Masing masing depot diasumsikan cukup besar untuk memenuhi permintaan beberapa *customer*. Dengan setiap *customer* hanya di kunjungi 1 kali pada setiap pemberangkatan. Surekha dan Sumathi (2011) dalam jurnalnya menyatakan bahwa pada dasarnya sasaran dari pada MDVRP adalah untuk meminimalkan jarak total pengiriman atau waktu yang dihabiskan dalam melayani semua *customer*. Sedikitnya waktu pengantaran, sebesar-besarnya kepuasan *customer*.



Gambar 1 di atas merupakan tahap penyelesaian MDVRP. Dimulai dengan pengelompokan, yang dimaksud pengelompokan di sini adalah mengelompokkan beberapa *customer* ke depot terdekat, diharapkan dengan pengelompokan ini dapat meminimalkan total jarak pengiriman dan menghindari pengiriman dari depot ke *customer* dengan jarak yang dalam perhitungannya kurang menguntungkan. Kemudian dilanjutkan dengan penentuan rute, penentuan rute ini tergantung berdasarkan metode yang digunakan. Setelah rute ditentukan, tahap terakhir adalah mengurutkan rute yang bertujuan untuk mengoptimalkan rute tersebut, pengurutan ini dimulai dengan *customer* terdekat dari depot dilanjutkan dengan *customer* berikutnya yang terdekat dengan *customer* sebelumnya, dan seterusnya.

Surekha dan Sumathi (2011) dalam jurnalnya menuliskan bahwa MDVRP merupakan Formulasi dengan tujuan dalam bentuk rangkaian dari *customer* dalam setiap rute kendaraan. Waktu yang diperlukan untuk berkeliling antar *customer* dengan depot dan permintaan yang diketahui. Diasumsikan bahwa semua kendaraan mempunyai kapasitas yang sama, dan setiap kendaraan mulai berkeliling dari depot, dalam penyelesaian dari pelayanan ke *customer*, dan kembali ke depot. Notasi yang digunakan dan model matematikanya adalah sebagai berikut:

Himpunan:

I = Himpunan dari semua depot

J = Himpunan dari semua *customer*

K = Himpunan dari semua kendaraan

Indeks:

i = indeks depot

j = indeks *customer*

k = indeks rute/kendaraan

Parameter:

N = Banyaknya kendaraan

C_{ij} = jarak antara titik i dan j

V_i = kapasitas maksimal pada depot i

$d_{\square j}$ = permintaan *customer* j

Q_k = kapasitas kendaraan (rute) k

Variabel keputusan:

$$x_{ijk} = \begin{cases} 1, & \text{jika kendaraan } k \text{ dijalankan dari titik } i \text{ ke } j \\ 0, & \text{untuk yang lainnya} \end{cases}$$

$$z_{ij} = \begin{cases} 1, & \text{jika customer } j \text{ dialokasikan ke depot } i \\ 0, & \text{untuk yang lainnya} \end{cases}$$

U_{lk} = pelengkap variabel untuk penyisihan batasan sub-tour pada rute k

Model matematika:

Fungsi tujuan adalah untuk meminimalkan jarak total pada semua kendaraan diberikan oleh persamaan (1)

$$\min \sum_{i \in I \cup J} \sum_{j \in I \cup J} \sum_{k \in K} C_{ij} x_{ijk}$$

Setiap customer telah dimuat dalam rute tunggal menurut persamaan (2)

$$\sum_{k \in K} \sum_{i \in I \cup J} x_{ijk} = 1, \quad j \in J$$

Batasan kapasitas untuk sebuah himpunan kendaraan diberikan oleh persamaan (3)

$$\sum_{j \in J} d_j \sum_{i \in I \cup J} x_{ijk} \leq Q_k, \quad k \in K$$

Persamaan (4) memberikan himpunan batasan penyisihan sub-tour baru sebagai berikut,

$$U_{lk} - U_{jk} + N x_{ijk} \leq N - 1, \quad l, j \in J, \quad k \in K$$

Setiap kendaraan harus meninggalkan outlet yang telah dikunjungi ditunjukkan pada persamaan (5)

$$\sum_{j \in I \cup J} x_{ijk} - \sum_{j \in I \cup J} x_{jik} = 0, \quad k \in K, \quad i \in I \cup J$$

Setiap rute dapat melayani sebanyak sekali berdasarkan persamaan (6)

$$\sum_{i \in I} \sum_{j \in J} x_{ijk} \leq 1, \quad k \in K$$

Batasan kapasitas untuk depot diberikan dalam persamaan (7) seperti,

$$\sum_{j \in J} d_j z_{ij} \leq V_i, \quad i \in I$$

Batasan dalam persamaan (8) menetapkan bahwa *customer* dapat ditempatkan pada sebuah depot hanya jika terdapat sebuah rute dari depot tersebut yang melewati *customer* tersebut.

$$-z_{ij} + \sum_{u \in I \cup J} x_{iuk} + x_{ujk} \leq 1, \quad i \in I, \quad j \in J, \quad k \in K$$

Syarat biner dalam variabel keputusan diberikan oleh persamaan (9) dan (10)

$$x_{ijk} \in \{0,1\} \quad i \in I, \quad j \in J, \quad k \in K$$

$$z_{ij} \in \{0,1\} \quad i \in I, \quad j \in J$$

Nilai positif pada variabel pelengkap didefinisikan dalam persamaan (8) seperti,

$$U_{lk} \geq 0, \quad l \in J, \quad k \in K$$

METODE UPPER BOUND PADA MDVRP

Berikut ini merupakan tahap-tahap penyelesaian Metode *Upper Bound* pada *Multi Depot Vehicle Routing Problem* (MDVRP):

Pengelompokan

Surekha dan Sumathi (2011) dalam jurnalnya menyatakan bahwa pada tahap ini setiap *customer* dikelompokkan berdasarkan depot terdekat. engelompokan ini dilakukan berdasarkan pada penghitungan jarak dengan aturan sebagai berikut:

- Jika $c_{i_1 d_1} < c_{i_1 d_2}$, maka outlet i_1 dikelompokkan pada depot d_1

- Jika $c_{i_1 d_1} > c_{i_1 d_2}$, maka outlet i_1 dikelompokkan pada depot d_2
- Jika $c_{i_1 d_1} = c_{i_1 d_2}$, maka outlet i_1 dikelompokkan pada sebarang depot antara d_1 dan d_2

Dengan:

$D = \{d_1, d_2, \dots, d_p\}$ adalah himpunan titik yang mewakili depot.

$I = \{i_1, i_2, \dots, i_q\}$ adalah himpunan titik yang mewakili outlet

$D \cup I = \{0, 1, 2, \dots, n\} = N$ adalah himpunan semua titik mewakili depot dan outlet.

$c_{i_1 d_1}$ adalah jarak antara outlet i_1 dengan depot d_1

$c_{i_1 d_2}$ adalah jarak antara outlet i_1 dengan depot d_2

Dalam hal ini outlet merupakan *customer*.

Pembentukan Rute

Pada tahap ini pembentukan rute dilakukan secara terpisah berdasarkan hasil pengelompokan yang telah dilakukan. Pencarian rute menggunakan Metode *Upper Bound* pada *Multi Depot Vehicle Routing Problem* (MDVRP). Adapun langkah-langkahnya sebagai berikut:

Dimulai dengan himpunan titik hingga, dimana setiap pasang titik dihubungkan oleh sisi berbobot.

Langkah 1: Pilih depot sebagai titik pertama dan temukan sebuah titik/*customer* yang terhubung oleh sebuah sisi berbobot minimum. Gambarlah dua titik tersebut dan hubungkan dengan dua sisi membentuk *cycle*. *Cycle* diorientasikan searah jarum jam.

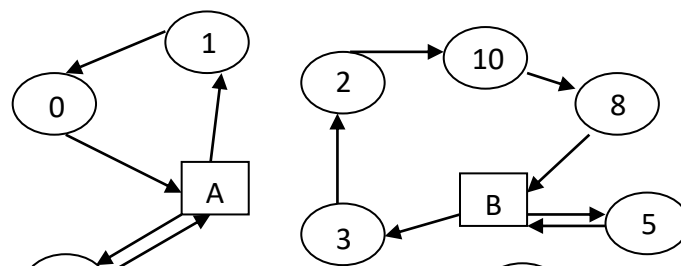
Langkah 2: Temukan sebuah titik yang sekarang tidak tergambar terhubung oleh sebuah sisi dengan bobot tersekit pada titik yang telah tergambar. Masukkan titik baru tersebut kedalam *cycle* di depan sisi terdekat yang telah terhubung.

Pemasukan titik baru pada langkah 2 tersebut akan terjadi apabila tidak melanggar kapasitas. Ulangi langkah 1 dan langkah 2 sampai semua titik termuat dalam rute.

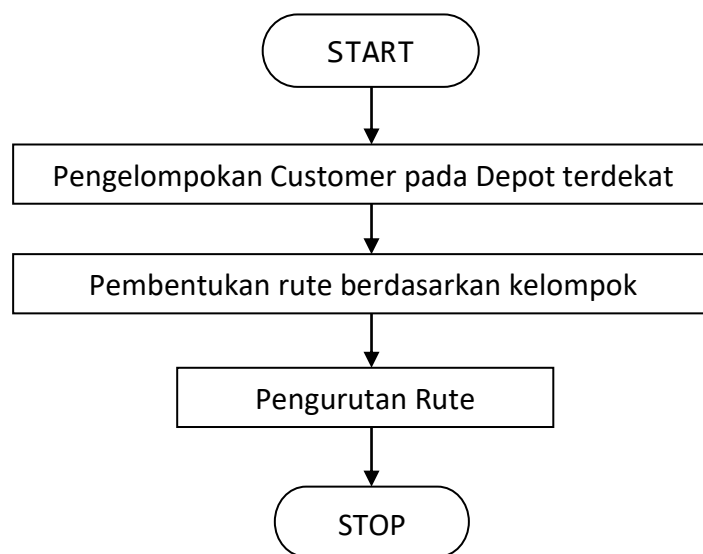
Pengurutan Rute

Pada tahap ini rute-rute yang telah terbentuk diurutkan berdasarkan jarak yang terdekat sehingga urutan pengiriman dimulai dari depot ke *customer* terdekat, kemudian ke *customer* berikutnya yang terdekat dari *customer* sebelumnya, dan seterusnya. Sehingga diperoleh rute-rute pengiriman dengan jarak terurut sebagai solusi yang mungkin dari *Multi Depot Vehicle Routing Problem* (MDVRP).

Hasil dari pencarian MDVRP merupakan beberapa rute yang berangkat dan kembali dari suatu depot. Tujuannya adalah meminimumkan rute dan jarak tempuh pendistribusian. Gambar 2 berikut ini merupakan gambaran dari suatu permasalahan MDVRP dengan 3 depot dan 12 *customer*, dimana *customer* 0,1,4 ditempatkan pada depot A, *customer* 2,3,5,8,10 ditempatkan pada depot B, dan *customer* 6,7,9,11 ditempatkan pada depot C.:



Berikut ini *block diagram* untuk metode *Upper Bound* pada *Multi Depot Vehicle Routing Problem* (MDVRP):



Gambar 3: *Block Diagram* metode *Upper Bound* pada MDVRP

CONTOH PERMASALAHAN

Berikut ini diberikan contoh permasalahan MDVRP yang dirujuk dari skripsi Masrurroh(2012) . Penggunaan contoh ini bertujuan untuk membandingkan dengan contoh yang telah dikerjakan sebelumnya.

Suatu perusahaan akan melakukan pengiriman barang kepada *customer* yang tersebar di suatu daerah. Perusahaan tersebut memiliki 2 depot yang terletak diantara *customer–customer*, yaitu depot 0 dan depot 1. Kapasitas tiap kendaraan sebesar 70 box. Adapun daftar permintaan tiap *customer* (dalam box) diberikan pada tabel berikut :

Tabel 1 : Jumlah Permintaan tiap *customer*

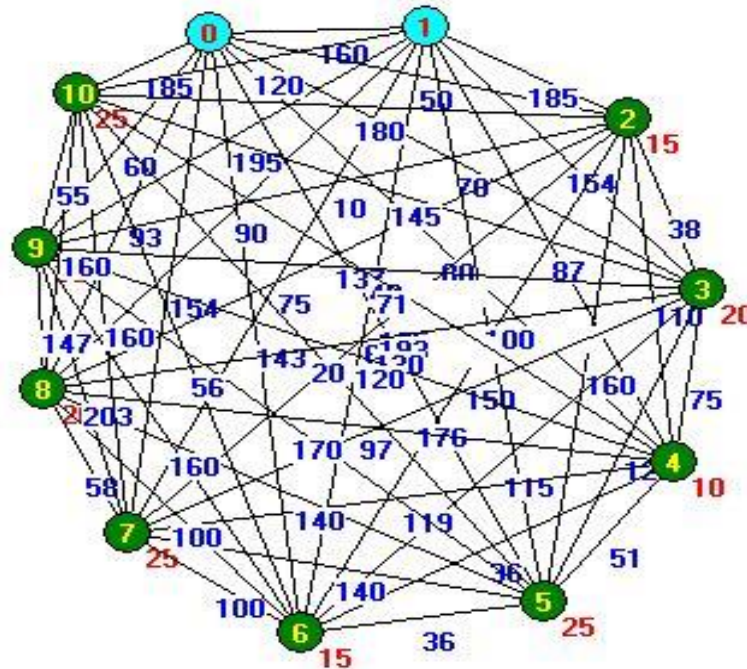
<i>Customer</i>	2	3	4	5	6	7	8	9	10
Permintaan	15	20	10	25	15	25	20	30	25

jarak antara titik, yaitu depot dengan *customer* (dalam satuan km) ditunjukkan pada tabel berikut:

Tabel 2 : jarak antar depot dengan *customer* dan antar *customer*

	0	1	2	3	4	5	6	7	8	9	10
0	0	160	50	70	80	165	143	154	93	60	185
1	160	0	185	154	87	100	60	75	90	195	120
2	50	185	0	38	110	160	150	193	137	10	180
3	70	154	38	0	75	125	115	176	130	48	145
4	80	87	110	75	0	51	36	119	97	120	71
5	165	100	160	125	51	0	36	140	140	170	20
6	143	60	150	115	36	36	0	100	100	160	56
7	154	75	193	176	119	140	100	0	58	203	160
8	93	90	137	130	97	140	100	58	0	147	160
9	60	195	10	48	120	170	160	203	147	0	55
10	185	120	180	145	71	20	56	160	160	55	0

Berdasarkan tabel di atas dapat dibentuk graph sebagai berikut, dimana pada C_{ij} untuk titik i dan j , i mewakili baris dan j mewakili kolom pada tabel. Sebagai contoh C_{20} berarti pada baris 2 dan kolom 0 berjarak 50 dan C_{21} berarti pada baris 2 kolom 1 berjarak 185.



Gambar 4: Graph Lengkap

Penyelesaian dengan Metode *Upper Bound* Pada *Multi Depot Vehicle Routing Problem* (MDVRP)

1. Pengelompokan

Berdasarkan data jarak dari tabel jarak, *customer* 2,3,4,dan 9 dikelompokkan ke dalam depot 0 dan *customer* 5,6,7,8, dan 10 dikelompokkan ke dalam depot 1.

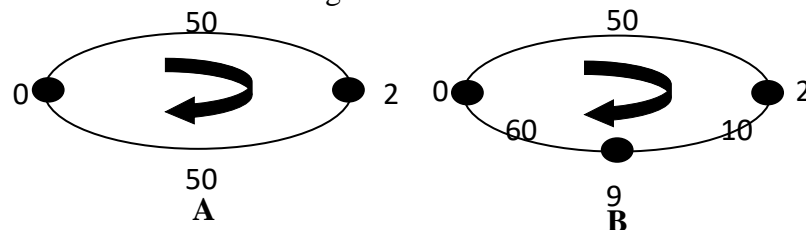
2. Pembentukan rute

Dimulai dengan himpunan titik hingga $G = \{0,2,3,4,9\}$ dan $H = \{1,5,6,7,8,10\}$

- Pembentukan rute untuk himpunan $G = \{0,2,3,4,9\}$

Langkah 1: **Rute 1**

Depot 0 sebagai titik pertama dan *customer* 2 sebagai titik terdekat. Kedua titik tersebut digambar dan dihubungkan dengan dua sisi sehingga membentuk sikel lihat gambar 5A.



Langkah 2: Pada gambar 5B *customer* terdekat dari depot 0 dan *customer* 2 adalah *customer* 9 (jarak 10 km) dengan kapasitas $15 + 30 = 45 < 70$. *Customer* 9 dimasukkan di depan *Customer* 2 di dalam *cycle*.

Pada gambar 5C *customer* terdekat dari depot 0, *customer* 2 dan *customer* 9 adalah *customer* 3 (jarak 38 km) dengan kapasitas $15 + 30 + 20 = 65 < 70$. *Customer* 3 dimasukkan di depan *Customer* 2 di dalam *cycle*.

Pada gambar 5D *customer* terdekat dari depot 0, *customer* 2, *customer* 9, dan *customer* 3 adalah *customer* 4 (jarak 75 km) dengan kapasitas $15 + 30 + 20 + 10 = 75 > 70$. *Customer* 4 telah melanggar kapasitas apabila dimasukkan ke dalam *cycle*.

Rule 2: Karena *customer* 4 merupakan titik terakhir yang belum termuat dalam rute, maka *customer* 4 terbentuk *cycle* dengan depot 0

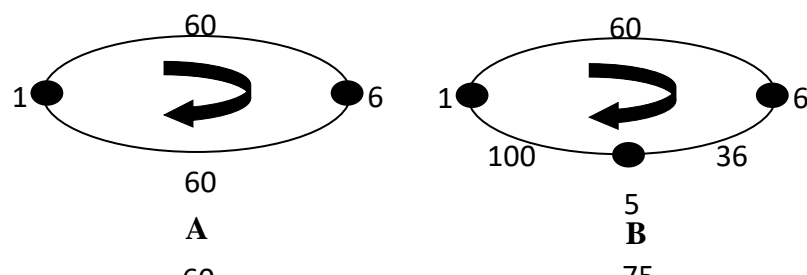
Semua titik dalam himpunan G telah termuat dalam rute sehingga diperoleh dua rute sebagai berikut:

- **Rule 1** : $[0 - 2 - 3 - 9 - 0]$ dengan jarak total 196 km
- **Rule 2** : $[0 - 4 - 0]$ dengan jarak total 160km

- Pembentukan rute untuk himpunan $H = \{1,5,6,7,8,10\}$

Langkah 1: **Rule 3**

Depot 1 sebagai titik pertama dan *customer* 6 sebagai titik terdekat. Kedua titik tersebut digambar dan dihubungkan dengan dua sisi sehingga membentuk sikel lihat gambar 6B.



Langkah 2: Pada gambar 6B *customer* terdekat dari depot 1 dan *customer* 6 adalah *customer* 5 (jarak 36 km) dengan kapasitas $25 + 15 = 40 < 70$. *Customer* 5 dimasukkan di depan *Customer* 6 di dalam *cycle*.

Pada gambar 6C *customer* terdekat dari depot 1, *customer* 6, dan *customer* 5 adalah *customer* 10 (jarak 20 km) dengan kapasitas $25 + 15 + 25 = 65 < 70$. *Customer* 10 dimasukkan di depan *Customer* 5 di dalam *cycle*.
customer terdekat dari depot 1, *customer* 6, *customer* 5, dan *customer* 10 adalah *customer* 7 (jarak 75 km) dengan kapasitas $15 + 25 + 25 + 25 = 90 > 70$. *Customer* 7 telah melanggar kapasitas apabila dimasukkan ke dalam *cycle*.

Karena terdapat *customer* 7 dan *customer* 8 yang belum termuat dalam rute, maka kembali ke langkah 1

Langkah 1: **Rute 4**

Depot 1 sebagai titik pertama dan *customer* 7 sebagai titik terdekat. Kedua titik tersebut digambar dan dihubungkan dengan dua sisi sehingga membentuk *cycle* lihat gambar 6D.

Langkah 2: Pada gambar 6E *customer* terdekat dari depot 1 dan *customer* 7 adalah *customer* 8 (jarak 58 km) dengan kapasitas $25 + 20 = 45 < 70$. *Customer* 8 dimasukkan di depan *Customer* 7 di dalam *cycle*.

Semua titik dalam himpunan H telah termuat dalam rute sehingga diperoleh dua rute sebagai berikut:

- **Rute 3** : [1 – 6 – 5 – 10 – 1] dengan jarak total 236 km
- **Rute 4** : [1 – 7 – 8 – 1] dengan jarak total 223km

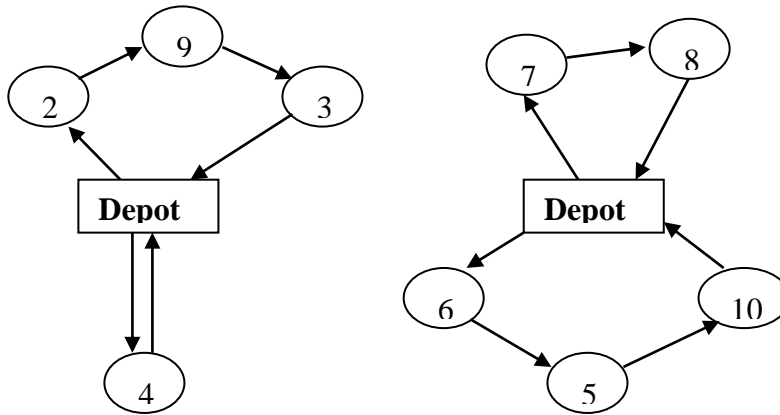
3. Pengurutan Rute

Pada tahap ini untuk masing-masing rute diurutkan dari *customer* terdekat depot, dan *customer* terdekat *customer* sebelumnya.

- **Rute 1** : [0 – 2 – 9 – 3 – 0] dengan jarak total 178 km

- **Route 2** :[0 – 4 – 0] dengan jarak total 160km
- **Route 3** :[1 – 6 – 5 – 10 – 1] dengan jarak total 236 km
- **Route 4** :[1 – 7 – 8 – 1] dengan jarak total 223km

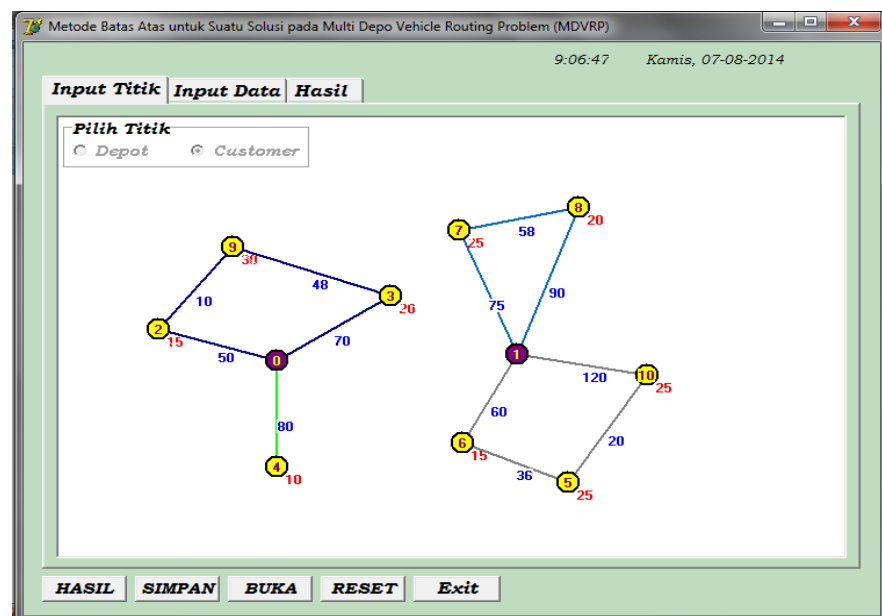
Gambar rangkaian rute yang telah terbentuk adalah sebagai berikut:



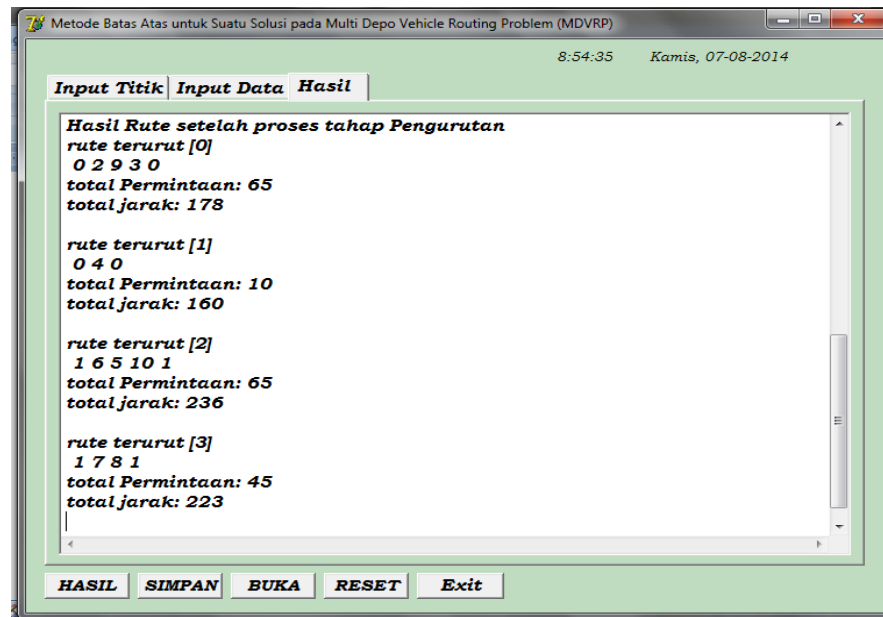
Gambar 7: Rangkaian rute

Pada skripsi Masruroh (2012), contoh permasalahan MDVRP di atas diselesaikan dengan menggunakan algoritma Clark and Wright. Solusi pencariannya yakni banyak rute 4 dan total jarak tempuh 797km. Pada skripsi Prihartinie (2013), contoh permasalahan MDVRP di atas diselesaikan dengan menggunakan metode *Insertion Heuristik*. Solusi pencariannya yakni banyak rute 4 dan total jarak tempuh 797km. Karena solusi yang dihasilkan sama, maka metode *Upper Bound* dapat digunakan untuk menyelesaikan masalah MDVRP.

Berikut ini merupakan tampilan program dan tampilan hasil penyelesaian contoh permasalahan menggunakan program:



Gambar 8: Tampilan visualisasi hasil



Gambar 9: Tampilan hasil penyelesaian menggunakan program

Berdasarkan gambar 8 dan 9 tersebut pada tab 'Input Titik' diperoleh gambar rute di setiap depot. Pada tab 'Hasil' diperoleh hasil berupa rute, yaitu:

- Rute [0] : 0 – 2 – 9 – 3 – 0, total permintaan 65, total jarak 178
- Rute [1] : 0 – 4 – 0, total permintaan 10, total jarak 160
- Rute [2] : 1 – 6 – 5 – 10 – 1, total permintaan 65, total jarak 236
- Rute [3] : 1 – 7 – 8 – 1, total permintaan 45, total jarak 223.

Program *Upper Bound* ini telah diuji coba dengan simulasi 7, 11, 21, 50, dan 100 titik.

KESIMPULAN

Berdasarkan uraian di atas, dapat dikemukakan beberapa kesimpulan sebagai berikut:

1. Metode *Upper Bound* dapat diterapkan pada permasalahan *Multi Depot Vehicle Routing Problem* (MDVRP). Tahap pertama adalah pengelompokan. Selanjutnya pada tahap Penentuan Rute, pada Metode *Upper Bound* dibentuk suatu himpunan titik berdasarkan pengelompokannya. Depot dipilih sebagai titik pertama yang terpilih, kemudian mencari *customer* dengan jarak terdekat untuk dihubungkan menjadi sebuah *cycle* searah jarum jam. Dilanjutkan dengan mencari *customer* dengan jarak terdekat dari *customer* dan depot yang terpilih yang tidak terdapat pada *cycle* tersebut. *Customer* baru yang terpilih akan disisipkan pada *cycle* apabila memenuhi kendala kapasitas kendaraan yang digunakan. *Customer* yang belum terpilih akan dibentuk *cycle* baru dengan depot yang sama dengan langkah- langkah yang sama. Setelah semua *customer* termuat didalam *cycle* proses berhenti. Tahap yang terakhir adalah Pengurutan rute.
2. Pada penerapannya Metode *Upper Bound* pada *Multi Depot Vehicle Routing Problem* (MDVRP) menghasilkan 4 rute dengan total jarak 797km.
3. Pencarian solusi Metode *Upper Bound* pada *Multi Depot Vehicle Routing Problem* (MDVRP) menggunakan bahasa pemrograman *Borland Delphi*. Terdapat beberapa tahapan dalam program ini diantaranya inputan titik yang berupa depot dan customer, inputan data berdasarkan tabel yang disediakan, dan hasil rute berupa rangkaian rute, total permintaan, total jarak, dan visualisasi hasil berupa graph dengan sisi-sisi yang menunjukkan suatu rute. Keabsahan hasil program tersebut dapat di lihat dari gambar 16 yang mana solusi pencarian dengan perhitungan secara manual dan program menghasilkan solusi yang sama.

SARAN

Bagi pengkaji ataupun peneliti selanjutnya diharapkan dapat:

1. Menggunakan program *Upper Bound* sebagai alternatif pencarian rute pada MDVRP.
2. Memodifikasi metode-metode lain dari *Travelling Salesman Problem*(TSP) pada MDVRP sehingga diharapkan dapat menemukan suatu solusi yang lebih minimum.

DAFTAR PUSTAKA

- Aldous, Joan M dan Wilson, Robin J. 2004. *Graph and Application an Introductory Approach*. London:Great Britain
- Caccetta, Louis, Mamoon Alameen, dan Mohamed Abdul-niby. 2013. *An Improved Clarke and Wright Algorithm to Solve the Capacitated Vehicle Routing Problem*. ETASR - Engineering, Technology & Applied Science Research, Vol. 3, _o. 2, 2013, 413-415.
- Campbell, Ann Melisa dan Martin Savelsbergh. 2004. *Efficient Insertion Heuristics for Vehicle Routing and Scheduling Problems*. Transportation Science, Vol. 38, No. 3, August 2004, pp. 369–378.
- Masruroh, Annisa. 2012. *Algoritma Clark and Wright pada Multi Depot Vehicle Routing Problem*. Malang: Universitas Negeri Malang.
- Prihartinie, Dima. 2013. *Penyelisaian Multi Depot Vehicle Routing Problem(MDVRP) menggunakan metode Insertion Heuristic*. Malang: Universitas Negeri Malang.
- Surekha dan Sumathi.2011. *Solution To Multi-Depot Vehicle Routing Problem Using Genetic Algorithms*. World Applied Programming, Vol (1), No (3), August 2011. 118-131.